Practice Problems – Divide-Conquer-Combine

1. Given an array of integers A, design and analyze a divide-conquer-combine based algorithm to find the maximum integer in the array.

2. In a 'Maximum Subarray Problem', given an array A of integers, the task is to find a contiguous subarray of A such that the sum of the values in the subarray is the maximum. Here, there is no restriction on the size of the subarray that has maximum sum or the nature of integers in the array or the subarray. Design and analyze an efficient algorithm for each of the following variants of the problem:
   a) Result is a contiguous subarray whose sum is the maximum and the length is minimal.
   b) Result is a contiguous subarray of non-negative integers whose sum is the maximum and the length is minimal.
   c) Input is an array of non-negative integers (zero inclusive). Result is contiguous subarray whose sum is the maximum and the length is minimal.
   d) Result is a non-contiguous subarray whose sum is the maximum.

3. Given an array A containing 'n' distinct numbers, an index pair (i, j) is called an inversion if $i < j$ and $A[i] > A[j]$. For example, the array $A = \{2, 3, 8, 6, 1\}$ contains 5 inversions: (1,5), (2, 5), (3, 4), (3, 5), (4, 5). Design and analyze a divide-conquer-combine based algorithm to compute the number of inversions in the array.

4. Given two square matrices of the same size, design and analyze a divide-conquer-combine based algorithm to compute the product of the two matrices.